VOL. 11, NO. 1, JANUARY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

SURVEY ON RELATIONAL DATABASE WATERMARKING TECHNIQUES

Abd. S. Alfagi[1], A. Abd. Manaf[1], B. A. Hamida[2], S. Khan[2] and Ali A. Elrowayati[3]
[1]Advanced Informatics School, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia
[2]Department of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur, Malaysia
[3]Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Malaysia

ABSTRACT

Digital watermarking has been in multimedia data use over the past years. Recently it has become applicable in relational database system not only to secure copyright ownership but also to ensure data contents integrity. Further, it is used in locating tampered and modified places. However, the watermarking relational database has its own requirements, challenges, attacks and limitations. This paper, surveys recent database watermarking techniques focusing on the importance of watermarking relational database, the difference between watermarking relational database and multimedia objects, the issues in watermarking relational database, type of attacks on watermarked database, classifications, distortion introduced and the embedded information. The comparative study shows that watermarking relational database can be an effective tool for copyright protection, tampered detection, and hacker tracing while maintaining the integrity of data contents. In addition, this study explores the current issues in watermarking relational database as well as the significant differences between watermarking multimedia data and relational database contents. Finally, it provides a classification of database watermarking techniques according to the way of selecting the candidate key attributes and tuples, distortion introduced and decoding methods used.

Keywords: database watermarking, digital watermarking, watermarking techniques, attacks, relational databases.

INTRODUCTION

Many applications provide a wide range of web-based services such as database, digital libraries, digital repositories, health services, and e-commerce, etc. These applications aim to make the digital assets secured, easily retrievable; making sure data is archived properly and shared electronically [1]. Consequently, more challenging issues of data piracy, tampering, copyrighting, ownership claiming, illegal redistribution, and data integrity checks arise. These issues are the main concern of data owners, though several security mechanisms have been deployed for database protection of access control and encryption. However, those security mechanisms protect the exposure of sensitive information before accessing that information [2]. Once the data is accessed the data is no longer protected against tampering, copyrighting, and illegal redistribution [3]. Therefore, digital watermarking techniques are studied here from viewpoint of deter piracy, tampering, copyrighting, ownership proof, and data integrity check [4-6].

DIGITAL WATERMARK

Digital watermark is a piece of information inserted into data for copyright protection, proof of ownership, traitor tracing, and integrity check. Generally, watermarking system consists of two phases, namely, watermark embedding and extraction as shown in Figure-1. In the embedding phase, a watermark information (W) is securely inserted into the original database using secret key (K). Then the watermarked database becomes ready for publication or distribution. The extraction phase is important to verify the originality and integrity of database. In this phase, the watermarked database is taken as input for extracting the watermark information using same key. The extracted watermark is compared with the original watermark information to verify the ownership of a suspicious database [7-9] or to protect the copyright [10-12]. A watermark detection process can be applied to any database so as to determine whether or not a legitimate watermark can be detected [13].

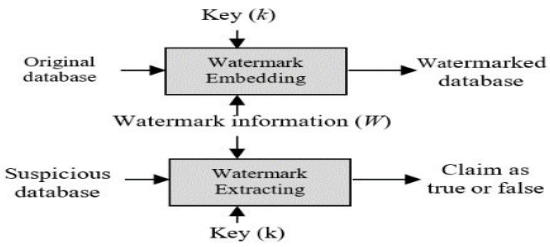

Figure-1. Basic framework for watermarking system [13].

IMPORTANCE OF WATERMARKING DATABASE

Digital watermarking is important in many real life applications for proof data ownership, protect copyright, fingerprinting data, and to preserve relational database integrity [15-17]. In ownership proof, data owners can securely insert a watermark into a relational database using secret key before publishing or distributing their data. At the extraction phase, the data owners can demonstrate the presence of their watermark in order to verify third party ownership claiming. Similarly, for copyright protection, data owners securely embed owner's specific information (e.g. image, text, speech etc.) into the relational database in order to prevent others from claiming copyright [18]. In the fingerprinting data, the

main goal of watermark is to identify the sources of data. In case, if the owner would choose the distinct watermarks within the content are supplied to different customers. Here, the watermark is helpful in identifying the illicit customers thus blocking supplying the content to unauthorized customers [15]. According to [17], [18] and [20] fragile watermark are useful for content authentication and integrity verification.

DIFFERENCES BETWEEN WATERMARKING MULTIMEDIA OBJECT AND WATERMARKING RELATIONAL DATABASE

Watermarking database is quite similar to the process of watermarking multimedia which has long been comprehensively studied [3, 13]. Table-1 illustrates significant differences between watermarking database and multimedia object. Therefore, watermarking relational database requires a class of information-hiding mechanism different to those used by multimedia watermarking. In addition, the nature of database relation is to accept changes like adding, deleting or modifying contents frequently. Thus, no existing watermarking techniques for image or audio are designed to accommodate such tuple operations [16, 22, 23]. These differences lead to increasing in research efforts as well as give a rise to many technical challenges in developing techniques for watermarking relational database.

Table-1. Differences between watermarking multimedia data and database.

According to	Multimedia data	Database
Redundant Data	Has large number of bits	Few
Embedding positions	Not changeable	Changeable
Updating	Data is not replaced	Frequently Updated adding, deleting and modifying
Human visual system	Psycho-physical phenomena can be exploited	Cannot exploit such phenomena
Size	Fixed	Changeable
Data type	Numerical data only	Numerical, categorical and non-numerical

DATABASE WATERMARKING ISSUES

Watermarking database is an important to deter traitors, tracing, fingerprints, locating tampered places, etc. Meanwhile, watermarking relational database not without many challenges [15]. The issues that arise in watermarking database are listed below:

- **Capacity:** The optimal amount of data that can be embedded in a carrier.
- **Usability:** The desirably embedded amount of watermark information into the database should not degrade the usability of data.
- **Robustness:** The embedded watermark should be robust to resist certain attacks whether they are malicious or accidental.
- **Blindness:** The knowledge needed to extract the watermark information from the watermarked database. The extraction should require neither the original un-watermarked database nor the watermark itself.
- **Structure:** Most databases are made of relational tables made from tuples (rows) and attribute (columns), and these relations may be inter-related to each other. As a result, once should consider that the attributes that have inter-relation or joined before watermarking process should not be altered during watermarking.
- **Security:** The security of watermarking system should rely on the primary keys (e.g. security key or choice of attributes and tuples) not on the algorithm. There keys should be kept secret and known to the database owner only. This point must achieve the requirements of public system where the security defense must lie only in the choice of the private parameters
- **Incremental watermarking:** When a database has been watermarked, the algorithm must compute the watermark values for only the added or modified tuples. The tuples that have not altered during watermarking process should not be re-watermarked.
- **Non-inference:** When there is a need to embed more than one watermark in a single database relation the embedded watermarks should not conflict with each other.
- **False Positiveness and false Negativeness:** false positive or false hit is the probability of a valid extraction of watermark from un-watermarked database. While, false negative or false miss is the probability of not detecting a valid watermark from watermarked database.

According to [13, 15] these are the most important issues that arise in watermarking relational database systems. However, as more techniques became available for watermarking relational database still they having one or more from those issues. For instance, some techniques have limited capacity, no usability constraint, have a high false positive rate, not suitable for database relations that need frequent updates, or not robust against simple attacks.

WATERMARKED DATABASE ATTACKS

A watermarked data may undergo certain types of attacks before it reaches the detector or extractor side. The attacks can be either intentional or unintentional leading to

destroy the watermarked data, removal of the watermark or addition of noise or extra information on the watermarked data [15]. Different types of attacks can be recognized depending on the type of the watermark as well as the application or the type of the data [24]. These attacks are categorized and described below:

1. **Benign updates:** it is the case when database contents are modified include adding (or deleting) new some tuples (or attributes) or modifying values of tuples.
2. **Value modification attack or malicious attacks:** this category of attack has the following attacks:

- **Bit attacks:** In this kind of attack, an attackers attempt to destroy the embedded watermark partially or totally by altering one or more bits in the watermarked tuples. Bit attack may be performed by: Randomly assigning values to certain bit positions which known as Randomization Attack, setting the bit positions to zero which known as zero attack or inverting some values of bits which known as bit flipping attack.
- **Rounding attack:** This kind of attack is conducted by rounding all or major values of the numerical attribute. The success of rounding attack depends on the guesstimate of how many bits have been involved from that attribute in the watermarking. Underestimation of bits number may cause the attack unsuccessful, whereas overestimation may cause the data useless. For example, rounding number like 612.689 to become 613 this may damage the watermark especially if the watermark bit has been embedded in the fractional part.
- **Transformation:** In this attack the numeric values are linearly transformed. For example, attackers may convert the data to a different unit of measurement (e.g., Centimeter to Meter, feet or inch, or Fahrenheit to Celsius).

3. **Subset attack:** This kind of attack just causes modification, deletion or addition on a subset of tuples. Consequently, the watermark may lost or effected.
4. **Superset attack:** In this attack, attackers attempt to add new tuples or attributes to the watermarked database for miss leading the correct detection of the embedded watermark.
5. **Collusion attack:** In this attack, multiple fingerprinted copies of identical relation are required to be accessed by the attackers. collusion attack may be performed by:-
   - **Mix and match attack:** In this case, attackers may create their relation by taking disjoint tuples from numerous relations having same information.
   - **Majority attack**: a new relation has to be created in this attack with the same schema as the watermarked copies. The main goal of such attack is to prevent the owner from detecting the watermark.
6. **False claim of ownership:** In this attack, attackers attempt to insert another watermark in order to conflict the merchant's claim. It includes Additive attack and inevitability Attack.
7. **Subset reverse order attack:** this attack is just exchanging the order or positions of the tuples or attributes in the relation which may erase or disturb the embedded watermark specially that depend on the order of tuples like fragile watermark.
8. **Brute force attack:** In this case, the attackers try to guess about the private parameters (e.g. secret key) by traversing the possible search**.**

According to [15, 16, 25, 26], these are the most common attacks which watermarked database may undergo. Researchers are working on providing a digital watermark that is able to resist as much kind of attacks as possible. However, [18] stated that, "Till to date only few types of attacks are overcome" and recommends the need for strengthening the watermark and increasing the level of attack resilience.

CLASSIFICATION OF DATABASE WATERMARKING TECHNIQUES

So far, most proposed database watermarking techniques can be classified as shown in Figure-2 along different categories [13, 15, 16].

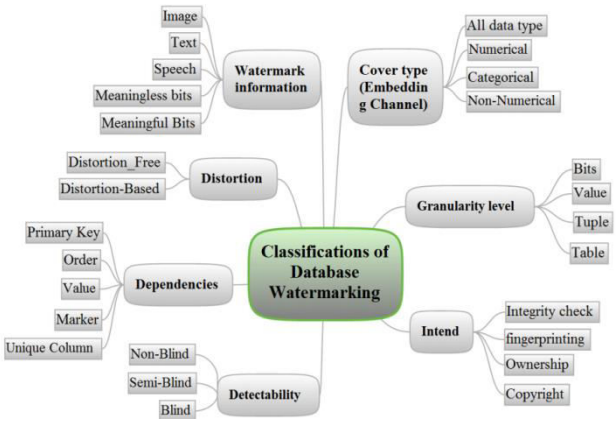

Figure-2. Classification of database watermark.

DATABASE WATERMARKING TECHNIQUES

Watermarking database techniques can be classified based on different factors such as watermark information, data type, introduced distortion, granularity level, detectability or intent. This section focuses on:

(i) Whether the watermark introduces any distortion,
(ii) Type of the underlying data and
(iii)Type of the embedded information

In addition, this section provides a comparison table for most reviewed techniques. Based on whether the watermark introduces any changes in the underlying data of the database, the watermarking techniques can be classified into two categories: 1) distortion-based and 2) distortion-free.

Distortion-Based database watermarking techniques

There are many database watermarking techniques introducing distortion to the underlying data such as [9, 14, 22, 24, 27, 28, 34-46]. The main assumption of these techniques is that the underlying data of the database can tolerate small amount of change during the embedding without affecting the usability of database. The watermarking can be performed at bit or character level, or higher such as attribute or tuple level, over the numerical attribute values [47]. Most of distortion-based techniques are intended for copyrights protection, and ownership proof. The distortion-based techniques can be categories based on the embedding information such as embedding meaningless, image, speech, etc. Table-2 illustrates and summarizes most distortion-based techniques.

Watermarking techniques with no specific watermark information

The AHK algorithm [27] technique is based on watermarking numerical attributes and the watermark embedding introduce a distortion into the underlying data using bit level by changing the values of least significant bits (*LSB*) of some arbitrary attributes. The main idea of this technique is to ensure that some bit positions for some of the watermarked attributes contain specific values in some tuples. These specific values constitute the existence of the watermark in the attribute consequently it proof that the watermark is existing in the relational database. Keys (γ, v, ξ and K) algorithmically determine the selection of attributes, tuples, and bit positions. The γ represent the number of selected tuples, v represent the number of attributes, ξ represents the number of least significant bits and k represent the secret key. These parameters kept secret and known to owner only. A tuple is selected for watermarking if the hash value of its primary key module γ is equal to zero. The HASH function $H(K||r.P)$ is used to securely determine the candidate bit values to be embedded at selected positions. This function applied to compute the tuple selection. If this hash value is even, j^{th} LSB of the attribute values is set to *0*; otherwise, it is set to *1*. The watermark detection algorithm starts by identifying marked tuples, the marked attribute and marked bit. This bit is matched with the embedded bit and a threshold τ for matching bit count is computed. If the match count is greater than or equal to this threshold, watermark detection is said to be successful. Similar work provided by [34, 48] for watermarking relational but instead of using HASH and MAC they use the pseudorandom sequence generator that seeded with secret key and primary key of the current tuple. The main assumption of Agrawal and Kiernan is that the relation has unique primary key with unchangeable value. The authors in [27, 34, 48] did not use any mechanism for data usability control and they do not account for multi-bit watermarks, which makes their technique vulnerable against simple attacks.

Qin [12] proposed copyright protection technique for numerical database. The idea of chaotic random numbers is used for embedding watermarks in order to overcome the shortcomings of [27, 34, 48]. Qin uses Logistic Chaos Equation instead of MAC and HASH functions. The LCE selection depends on the tuple's primary key, which determines whether, where and what kind of the watermark is to be embedded. LCE has two importance properties compared to MAC and HASH functions. Firstly, LCE is a non-repetitive iterative operation, and secondly, its sensitiveness to the initial value [15]. Those features avoid the inherent weakness of collision that may occur by using Hash function. The MAC and HASH function simply use the same ξ Least Significant available for watermark insertion. Therefore, the error caused by watermark is decreased significantly and hardly affects the availability of the database. In the Qin system, watermark is inserted into the database under the control of the secret key and the chaotic random numbers. In addition, it is blind and the copyright is judged by the match rate of watermark as in [27, 34, 48].

Gupta [43] work to protect a database copyright and solve the issue of irreversibility. Gupta uses Deference Expansion (DE) technique. Two numeric attributes are selected from the same tuple of a database relation and a particular watermark bit is embedded into LSB of the selected attributes. In the watermark detection phase, difference between the maximum and minimum attribute value and its reverse value is calculated to compare them with the marked bits. Finally, the percentage of matching bits is calculated. However, this technique was vulnerable to secondary watermark attack and alteration attacks.

In another attempt, [42] try to overcome the mentioned attack by modified version of [27]. In the modified scheme, the original un-watermarked version of the relational database can be recovered along with the ownership proof by extracting the *Bit* from the integer portion of the attribute value before replacing it by the watermark bit and inserts it in the fraction portion of the attribute value. Both [43] and [42] encourage the owner to distribute a trial version of the database, which can only be recovered by users who only have purchased the key. While [42] solved the problem of secondary watermarking attack with a considerable distortion of the underlying data.

Xiao, Sun [52] exploited second least significant bits to minimize the distortion. The watermark bits are randomly generated in range between *0* and *9*. In the embedding phase, the tuples algorithmically selected with secure key and for each tuple, a hash value i is computed as $i = H(K_2||r.P) \bmod 10$. This hash value is then compared with the second LSB of the tuple and if this bit is *1*, the LSB is set to a pseudo random integer between *0* and *9*; otherwise, no change is done. The decision whether to mark i^{th} ($1 \le i \le m$) group depend on the i^{th} bit of the owner's watermark. Whereas the selection of the tuples in a group is based on a secret key (which is different from that used during partitioning) as well as the information at second LSB positions of the numeric candidate attributes. Finally, random numbers (between *0* and *9*) for the

selected tuples are embedded at LSB positions in the attribute values of those tuples. In the watermark extraction phase, data is again grouped and the hash values of all tuples in each group are calculated and compared with the second LSB of the tuple if both values are same, the bit is decoded and a threshold based error correction mechanism is applied to finally decode the watermark. However, this technique is vulnerable to randomization attack and uncontrolled data alterations.

Xian [37] works for proof the ownership of relational database by embedding meaningless information. The data is watermarked with a secret watermark key and a secret shadowed key, which is a unique for every user. A trusted watermark server (TWS) watermarks the data and assigns the shadowed key to the data user. When illegal data leakage is detected, the shadowed key of the data user is used to identify the actual source of data leakage; as a result the innocent user is not falsely accused of data leakage. As in most of the watermarking techniques, this technique also marks the selected LSBs of the selected tuple after generating a sequence of pseudo-random numbers. In watermark detection phase, the sequence of pseudo-random numbers is again generated and marked bits are detected. A counter for matching bits is incremented by *1* for every correctly decoded bit. This counter should reach a predefined threshold for successful watermark decoding. However when the embedding algorithm faces Null value in the fractional attribute it terminates the loop and revokes the loop from continues.

Manjula [18] highlighted the problem of additive attacks and proposed a numerical database watermarking technique for copyright protection. This technique is based on partitioning the attributes into groups then selects all tuples to be watermarked. The main objective of selecting all tuples to be watermarked is to increase the collisions of watermark. The assumption is that, if overlapping regions occur so that it becomes very difficult for attacker to avoid collusion by launching an additive attack. The proposed watermarking scheme is quite simple which partitions the data into m logical groups. Then for each partition a watermark bit is embedded such that $k \bmod j = i$; where k is a partition number and j is the length of watermark. Therefore, this technique suggested that Alice should mark large fraction of tuples so that it becomes very difficult for Mallory to avoid collusion.

A technique that utilizes weights attribute for watermark embedding and detection is proposed by Hamadou, Sun [54]. They reorder all attributes in a secret way by calculating the hash values of attributes using a secret key and the attribute names. Each attribute is assigned a weight based on its importance for the data owner then cluster the attributes based on their weight from the less to the highest. The sensitive attributes are assigned high weights, semi-sensitive attributes are assigned intermediate weights and insensitive attributes are assigned low weights. The main assumption is that the attacker would not change the name of the attributes. Therefore, this method is vulnerable to some attack that targeted the attribute name since any change in the attributes name make the watermark undetectable.

Watermarking techniques with image as watermark information

Zhang [7] uses an image to watermark numerical database in order to proof the database's ownership. This method embeds the watermark bits into randomly selected tuples that have been specified from the numerical attributes as in [27]. Zhang scheme is based on the patchwork algorithm and the character of some numerical attributes values that are able to tolerate small change without effecting database usability. In the embedding phase, the algorithm reads the values of each pixel's RGB from the ownership image. The tuples are divided into subsets using secure function. Then a subset of tuples is selected for watermarking. The ordered *RGB* values of the image are embedded as a watermark in the database relation. At the detection phase, the same hash function is applied to identify the watermarked tuples and then the embedded pixel values are decoded.

In [56], authors used a binary image of BMP format as a watermark for watermarking numerical relation. The image is divided into two parts: the header and the image information segment or the data array. Both of them are stored for later use. the hash value for each tuple using the primary key concatenating with the formerly extracted header of the BMP is calculated. In the extraction phase, the watermark detection starts by calculating the hash values for each tuple in the database relation R then sorting the tuples based on their hash values. The *maskbit* bit is calculated, and an *XOR* operation is applied to the j^{th} bit of the marked attribute and the calculated *maskbit*. An interesting analysis presented by the authors is the resilience of proposed technique again additive-attacks.

The authors in [57] applied Arnold transform to generate a scrambled disordered image. The image is converted to a bit string of specific length to serve as a watermark bits. Then the data set is partitioned into non-overlapping groups based on the length of the watermark. The selection of tuples for watermarking is conducted same as [27]. After this step, the i^{th} watermark bit is embedded in the i^{th} group. In the watermark detection phase, the data is grouped, the marked tuples are located and the position of marked bits is computed. Applying the hash function on the primary key value of the watermarked tuple and the order of the image. Then compute the modulo of the hash value. In the same fashion, modulo of the hash value and the number of bits available for watermarking is computed and stored in a parameter j. the detected bit is *1* when the modulo is *1*. Otherwise, the detected bit is zero. In such a technique tuple's deletion, insertion and alteration attacks effects the watermark decoding accuracy.

Farfoura *et al* [9] presented a reversible method called prediction-error expansion for watermarking

relational databases. In this technique, an identification image is converted into bits stream then embedded into the fractional part of the numeric attribute. A one-way hash function is applied on the primary key and the secret key *K* to select the tuples for watermarking. The secret key is computed using the relation: $K = H(ID||DB_{name}||version||...)$; where, *ID* is the identity of the database owner, || means a concatenation operator, DB_{name} is the name of the database, *version* is the database version and *H*() is a cryptographic hash function.

Recently, Farfoura [22] provided a numerical database watermark technique for copyright protection. by inserting an identification image into the R. The image firstly converted into a bit flow then these bits embedded into the fraction portion of the candidate numeric attribute. *Get_int()* function is used to extract the integer part of a real number while *Get_frac()* is used to retrieve the fraction portion. In the extraction phase, the fraction portion processed by subtracting the fraction portion from *H(ti.P || K)* to calculate the expanded difference, and the watermark bit *1* is the LSB of the expanded difference. For each watermarked bit, they count the numbers of *0s* and *1s*, respectively. Then the higher value represents the final value of this bit. Since the detected result is a binary sequence, they have to transform the obtained binary sequence into the watermark image. Then the copyright holder can make use of the detected watermark image to prove the copyright ownership.

Watermarking techniques with speech as watermark information

Wang[31] uses owner's speech to for watermarking relational databases. In the embedding phase, the watermark is changed to a bit flow of a specific size by converting the speech signal into a signal of 8-bits size. The tuples are selected for watermarking by applying the hash function. The value of the selected LSB is equivalent to the output of *XOR* operation on the j^{th} watermark bit and the real bit of the attribute. In the watermark detection phase, a hash function is applied to find the watermarked tuple. Then the watermark is extracted by applying the *XOR* operation again on the located bit and the j_{th} watermark bit. After detecting all watermarks a majority voting scheme is used as correction mechanism. Another technique based on speech signals by Zhang [32]. They followed same algorithmic steps as in image-based technique proposed by [58].

A recent database watermarking technique based on speech provided by Zhao [33]. The speech signal and data owner's information are embedded into numerical relation for copyright protection. In the embedding phase, both signals are compressed using wavelet. Then, converting the obtained signal into 8-bit signal, the pixels arranged from left to right and from top to town to get a set of *S=(S1,S2,.....Sn)*. The embedding locations are chosen using one-way hash function. The extraction process is almost same as that suggested in [31]. However, using speech for watermarking database need several preparation steps such as compression, remove the noise and converte speech signal into bit stream

Watermarking techniques with content characteristics as watermark information

Zhang, Niu [10], provided a numerical database watermarking method for copyright protection. Some bits called local characteristic from the attribute *A1* of tuple *t* were extracted and embeded into the attribute *A2* of the same tuple. The selection of tuples depends on whether the generated random value (between 0 and 1) is less than the embedded proportion α of the relational databases and the non-null requirement of characteristic attributes value. At the detection of watermark phase they following similar procedure, the local characteristic of the attributes are extracted and compared.

A fragile watermarking scheme for verify the integrity of database relation proposed by Guo [38]. The technique securely divided all tuples in the relation into groups. Then two watermarks are generated *W1* and *W2* by extracting the sequence of hash values' bit. *W1* is formed according to the MAC and the same attribute of all tuples in the same group whereas, *W*2 formed from the same MAC and all attributes of the same tuple. *W1* is embedded in the LSB level, while *W2* is embedded at next to the LSB level. The embedded watermarks form a watermark grid, which help in detect, localize and characterize suspicion modifications.

Other meaningful information as a watermark

In Huang, Cao [59], the authors aim to embed a meaningful information into a numerical relational database. The idea is based on tuples *ID* that must be a unique. This *ID* is important to sort all tuples in ascending order according to their *ID* values. All tuples are partitioned into groups *p* and each group contains *m* tuples. The partitioning of tuples in most of the techniques is based on one-way hash function. The i^{th} bit of the bit flow is embedded into the selected tuples in i^{th} group. A usability constraint function is used to preserve the data usability and prevent change of tuple's value that exceeds the data usability constraint. A similar scheme by Huang [60] which use *k*-means algorithm to cluster the tuples into several same equivalent classes instead of partitioning. The technique of *k*-means assures that the locations of the embedded the watermark is irregular.

A generalized and adaptive relational data watermarking framework (GARWM) proposed by [61] that exploits the properties of relational data in the semantics of watermarking, such as preservation of logical relationship in usability preserving attack, discrimination in significance of attributes, and local constraints/global metrics, to strengthen existing methods. This framework including three phases; select the candidate groups from all attributes and record them as the watermark, Append the error correction code (ECC) to the watermark, and the insertion algorithm.

Table-2. Comparison of distortion-based watermarking techniques for numerical database.

Authors	Selecting attributes	Selecting tuples	Distortion level	Decoding method	Intend	Introduce distortion
Watermarking techniques with no specific watermark information						
Agrawal , Kiernan [27]	Arbitrary	SHF	LSB	Blind	Copyright	Yes
Agrawal, Haas [48]	Arbitrary	PRSG	LSB	Blind	Copyrights	Yes
Kiernan, Agrawal [34]	Arbitrary	PRSG	LSB	Blind	Copyrights	Yes
Qin, Ying [12]	All	Arbitrary	LSB	Blind	Copyright	Yes
Xiao, Sun [52]	Arbitrary	SHF	2nd LSB	Blind	Ownership	Yes
Gupta , Pieprzyk [43]	Arbitrary	Arbitrary	LSB	Not Blind	Copyrights	Yes
Gupta , Pieprzyk [42]	Arbitrary	SHF	LSB	Not Blind	Ownership	Yes
Xian and Feng [37]	All	Arbitrary	LSB	Not Blind	Ownership	Yes
Manjula [18]	Arbitrary	all	LSB	-	Ownership	Yes
Watermarking techniques with Image as watermark information.						
Zhang, Niu [7]	Arbitrary	Arbitrary	Value	Not Blind	Ownership	Yes
Zhou, Huang [56]	Arbitrary	SHF	LSB	Semi-Blind	Copyrights	Yes
Wang, Wang [57]	Arbitrary	SHF	LSB	-	Copyright	Yes
Hu, Cao [58]	Arbitrary	Arbitrary	LSB	Blind	Copyrights	Yes
Farfoura and Horng [9]	SHF	SHF	LSB	Blind	Ownership	Yes
Farfoura, Horng [24]	SHF	SHF	LSB	Blind	Copyright	Yes
Watermarking techniques with Speech as watermark information						
Wang, Cui [46]	Arbitrary	SHF	LSB	Blind	Copyright	Yes
Zhang, Gao [32]	Arbitrary	SHF	LSB	-	Copyright	Yes
Watermarking techniques with Content characteristics as watermark information						
Zhang, Niu [10]	Arbitrary	Random	value	Blind	Copyright	Yes
Guo, Li [38]	SHF	SHF	LSB	Blind	Data Integrity	Yes
Other meaningful watermark information						
Huang, Cao [59]	Arbitrary	All	Bit	-	Copyright	Yes
Huang, Yue [60]	Arbitrary	SHF	LSB	-	Copyright	Yes
Hu, Chen [61]	All	SHF	LSB	-	Data Integrity	Yes
Zhao, Li [33]	Arbitrary	SHF	LSB	Blind	Copyright	Yes

PRSG pseudorandom sequence generators.
SHF Secure hash function
CPSG Cryptographic pseudorandom sequence generators

Distortion-Free watermarking techniques

Techniques under this category do not introduce any distortion to the underlying data. Table-3 illustrates and summarizes most distortion-free techniques.

Extracting hash values as a watermark information

In this technique, the hash values for the tuple and the group are computed and serve as a watermark. Authors in [62] and [3] proposed a fragile watermarking schemes for detecting modifications. These schemes aimed to preserve the integrity of categorical databases that cannot tolerate any distortion. In [62] the tuples is partitioned based on the hash value. The secret parameters as primary key and secret key are the main input of hash function. The watermark of length equal to the number of tuple pairs in the group is extracted from the group level hash value and for every tuple pair. The order of each two

tuples are changed or unchanged according to their tuple hash values and the corresponding watermark bit. Meanwhile, the partitioning in [3] is based on categorical attribute values. The group size is very important parameter; if an attacker knows the group size that shared with the information of the secret key *K*, attacker can simply delete all tuples in a group as a result the scheme will fail to detect modification. In addition, the length of each watermark is *v/2*, which is the half of the group size this means the larger the group size, the larger the probability of discovering modifications but less accuracy in localizing modifications because there are more tuples in each group. Further, in these schemes any modification of an attribute value will affect the watermarks in two groups as the modified tuple may remove from one group and be added to the other group.

Combining owner's watermark and database features as a watermark information

Since database feature is content dependent, this feature is exploited by [14] to generate invariant database feature. The main idea is to generate a watermark *W* which is a white image of size $\sqrt{n} \times \sqrt{n}$ where *n* represents the number of tuples. For each tuple t_i in the relation a value C_i is created using hash function MD5 and XOR operation, where $0 \leq Ci \leq 255$. The feature *C* of length *n* is a combination of all C_i. The final code *R* is formed by XOR-ing *C* and *W*. After that, *R* is encrypted using private key *k* that is known to the database owner only. Then the encrypted form of *R* made available publicly. To check the integrity same steps are needed to generate similar image from the suspicious database.

Converting the database relation into binary image as watermark information

Bhattacharya and Cortesi [63] provide a method to strengthen the database watermark integrity for all type of database. It is based on grouping all tuples then generates a gray scale image from the grouped tuples in order to serves as tamper detection procedure. The data partitioning algorithm partitions the data set based on a secret key into non overlapping partitions, $[S_0],...,[S_{m-1}]$, such that each partition $[S_i]$ contains on average (η /m) tuples from the data set D. A static number of MSBs and LSBs of the particular field are used for creating the watermark of that corresponding field. The watermark value is a combination of *m* which is represented MSBs and *n* that indicate the LSBs. At the verification process, the secret key and watermark are needed to check the suspicious database.

A similar scheme by [64] proposed for numerical and non-numerical database. The main intend is to strengthen the verification of integrity of the relational databases. Bhattacharya uses a zero distortion authentication mechanism based on the abstract interpretation framework. The proposed watermarking technique is partition based. The main idea is to generate a binary image as in [65] or grey scale as in [63]. However, the discussed schemes are based on tuples and attribute positions if their positions changed by the attackers the watermark cannot be discovered correctly. In addition, a synchronization issue is the most common problem in converting database relations into binary images.

Watermarking techniques based on tuple or attribute insertion

Prasannakumari [66] proposes database watermarking scheme that does not modify the original contents of database data rather they may change the structure of the original database by adding extra columns for checksum bits. The values of the added column are calculated by gathering the numeric columns presented in the database relation. The authors suggest locking the added column using a secret key to ensure watermark security. Another method by [67] based on adding user fingerprint as extra column to recognize the authorized from unauthorized users. This scheme was abile to know the latest changes made by each user in the database. However, adding column needs to be locked using private and the watermark information may be lost by dropping only one attribute from the database relation without loss of data usability. In addition, using such schemes, the structure of the database will be changed by adding a new column. Defiantly, Changing database schema by adding new column need special knowledge and will increase storage and maintenance overhead.

Instead of adding extra column authors in [68] suggested to add fake tuples as a watermark. The fake tuple creation algorithm takes care of candidate key attributes and sensitivity level of non-candidate attributes. Authors use Bernoulli sampling probability p_i for the i^{th} non-candidate attribute A_i to decide its fake value, which may be chosen uniformly, or as the value with higher occurrence frequency in the existing set of values of A_i in the relation. Unlike other algorithms, the detection algorithm is not an inverse algorithm to the watermark generation algorithm and insertion algorithm is probabilistic in nature. Detection algorithm checks to see whether the fake tuples inserted during watermark [16]. In addition, this scheme is highly vulnerable to tuple deletion attacks because deletion of fake tuples will result in loss of watermark information [53].

Table-3. Comparison of distortion-free watermarking techniques.

Authors	Watermark information	Granularity level	Attribute type	Introduce distortion	Verifiability	Intent
Bhattacharya and Cortesi [3], [64, 65]	Generate a Binary Image	Group	Categorical	No	Blind public or private	Ownership or temper detection
Li, Guo [62]	Hash Value	Tuples	Categorical	No	Blind	Temper detection
Prasannakumari [66]	Adding Extra Columns	Checksum bits	Character data type	No	-	Data integrity
Zawawi, El-Gohary [67]	Adding Extra Columns	Fingerprint for each row	Categorical and numerical	No	-	Integrity And Copyright
Pournaghshband [68]	Generate Fake Tuples	Add a fake tuples decided by the database owner	All	No	Not Blind	Copyright
Bhattacharya and Cortesi [64]	Abstract Interpretation Framework	group based	All	No	Not Blind	Data integrity

CONCLUSIONS

In this paper, most recent numerical database watermarking techniques have been reviewed. The categorization of reviewed schemes is manly based on: (i) whether the schemes introduce a distortion to database contents named a distortion-based, and (ii) whether schemes introduce a zero distortion into the underlying data named a Distortion- free. In addition, schemes that categorized under the first category have been studied in depth and classified into six type-based on the type of embedded information, namely: techniques with specific, no specific, image, speech, and cloud data, meaningful data and with other information as watermark.

Meanwhile, in the second category the classification is differ and based on whether the schemes extract the hash values, combining owner's data with some database features and converting relational database into binary image or whether the schemes based on adding extra attribute or tuple to serve as a watermark. The distortion-based and distortion- free techniques with their subcategory are summarized in table1 and Table 2 respectively. Mostly the distortion-based techniques are a robust and used for copyrights protection or for ownership proof, whereas most distortion-free techniques are fragile and used for maintaining database integrity. In addition, most of distortion-based techniques follow almost similar steps to select the candidate bit positions for watermark embedding.

ACKNOWLEDGEMENTS

The authors would like to express greatest appreciation to Advanced Informatics School (AIS), Universiti Teknologi Malaysia (UTM) for financial support. Furthermore, thanks to a financial support by Ministry of Defense, Libya under their scholarship program for PhD studies of the first author.

REFERENCES

[1] Al-Sayid, N.A. and D. Aldlaeen, Database Security: Threats A Survey Study. International Conference on Computer Science and Information Technology (CSIT), 2013.

[2] Franco-Contreras, J., *et al.,* Authenticity Control of Relational Databases by Means of Lossless Watermarking Based on Circular Histogram Modulation, in Security and Trust Management. 2013, Springer. pp. 207-222.

[3] Bhattacharya, S. and A. Cortesi. A Distortion Free Watermark Framework for Relational Databases. in ICSOFT (2). 2009.

[4] Dadkhah, S., *et al*., An effective SVD-based image tampering detection and self-recovery using active watermarking. Signal Processing: Image Communication, 2014. 29(10): 1197-1210.

[5] Arun, R., *et al*. A Distortion Free Relational Database Watermarking Using Patch Work Method. In: Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012. Springer.

[6] Li, Z., J. Liu, and W. Tao, A Novel Relational Database Watermarking Algorithm Based on Clustering and Polar Angle Expansion. International Journal of Security and Its Applications, 2013. 7(2).

[7] Zhang, Y., *et al.* A Method of Verifying Relational Databases Ownership with Image Watermark. in The 6th International Symposium on Test and Measurement, Dalian, PR China, 2005.

[8] Shehab, M., E. Bertino, and A. Ghafoor, Watermarking Relational Databases Using Optimization-Based Techniques. IEEE Transactions on Knowledge and Data Engineering, 2008. 20(1).

[9] Farfoura, M.E. and S.-J. Horng, A Novel Blind Reversible Method for Watermarking Relational Databases. International Symposium on Parallel and Distributed Processing with Applications, 2010, pp. 563-569.

[10] Zhang, Y., *et al*. Relational databases watermark technique based on content characteristic. in Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on. 2006. IEEE.

[11] Zhou, M., *et al*. A Novel Fingerprinting Architecture for Relational Data. In: Digital EcoSystems and Technologies Conference, 2007. DEST'07. Inaugural IEEE-IES. 2007. IEEE.

[12] Qin, Z., et al. Watermark based copyright protection of outsourced database. In: Database Engineering and Applications Symposium, 2006. IDEAS'06. 10th International Conference. 2006. IEEE.

[13] Li, Y., Database Watermarking: A Systematic View, in Handbook of Database Security. 2008, Springer. p. 329-355.

[14] Tsai, M.-H., H.-Y. Tseng and C.-Y. Lai. A Database Watermarking Technique for Temper Detection. in JCIS. 2006.

[15] Halder, R., S. Pal, and A. Cortesi, Watermarking Techniques for Relational Databases: Survey, Classification and Comparison. J. UCS, 2010. 16(21): 3164-3190.

[16] Arathi, C., Literature Survey on Distortion based Watermarking Techniques for Databases. International Journal of Computer Science and Communication Networks. 2012. 2(4).

[17] Abdullah, S.M., A.A. Manaf, and M. Zamani, Capacity and quality improvement in reversible image watermarking approach, in Networked Computing and Advanced Information Management (NCM), 2010 Sixth International Conference on. 2010: Seoul. pp. 81 - 85.

[18] Manjula, R. and N. Settipalli, A new relational watermarking scheme resilient to additive attacks. International Journal of Computer Applications, 2010. 10(5): 1-7.

[19] khataeimaragheh, H. and H. Rashidi, A Novel Watermarking Scheme for Detecting and Recovering Distotions in Database Tables. International Journal of Database Management Systems, 2010. 2(3): 1-11.

[20] Khan, A., *et al*., A recent survey of reversible watermarking techniques. Information Sciences, 2014. 279: 251-272.

[21] Khan, A. and S.A. Husain, A fragile zero watermarking scheme to detect and characterize malicious modifications in database relations. ScientificWorldJournal, 2013. 2013: 796726.

[22] Farfoura, M.E., S.-J. Horng, and X. Wang, A novel blind reversible method for watermarking relational databases. Journal of the Chinese Institute of Engineers, 2014. 36(1): 87-97.

[23] Li, Y., H. Guo, and S. Wang, A multiple-bits watermark for relational data. Journal of Database Management, 2008. 19(3).

[24] Farfoura, M.E., *et al*., A blind reversible method for watermarking relational databases based on a time-stamping protocol. Expert Systems with Applications, 2012. 39(3): 3185-3196.

[25] Bilapatte, S., S. Bhattacharya, And S. Sawarkar, A Review On Watermarking Relational Databases. International Journal of Applied Engineering Research and Development (Ijaerd), 2014.

[26] S.Kshatriya, M.S. and Prof.Dr.S.S.Sane, A Study of Watermarking Relational Databases. International Journal of Application or Innovation in Engineering & Management (IJAIEM), 2014. 3(10).

[27] Agrawal, R. and J. Kiernan. Watermarking Relational Databases. In: VLDB Conference. 2002. Hong Kong,China,.

[28] Sion, R. Proving ownership over categorical data. in Data Engineering, 2004. Proceedings. 20th International Conference on. 2004. IEEE.

[29] Al-Haj, A. and A. Odeh, Robust and blind watermarking of relational database systems. Journal of Computer Science, 2008. 4(12): 1024.

[30] Al-Haj, A., A. Odeh, and S. Masadeh, Copyright Protection of Relational Database Systems, in Networked Digital Technologies, Pt 1, F. Zavoral, *et al*., (Editors). 2010. pp. 143-150.

[31] Wang, H., X. Cui, and Z. Cao. A Speech Based Algorithm for Watermarking Relational Databases. in ISIP. 2008.

[32] Zhang, Y.H., Z.X. Gao, and D.X. Yu, Speech Algorithm for Watermarking Relational Databases Based on Weighted. Advanced Materials Research, 2010. 121-122: 399-404.

[33] Zhao, X., L. Li, and Q. Wu, A Novel Multiple Watch marking for Relational Databases using Multi-Media. Physics Procedia, 2012. 25: 687-692.

[34] Kiernan, J., R. Agrawal, and P.J. Haas, Watermarking relational data: framework, algorithms and analysis. The VLDB Journal the International Journal on Very Large Data Bases, 2003. 12(2): 157-169.

[35] Sion, R., Rights assessment for relational data, in Secure Data Management in Decentralized Systems. 2007, Springer. pp. 427-457.

[36] Sion, R., M.J. Atallah, and S. Prabhakar, Rights protection for categorical data. Knowledge and Data Engineering, IEEE Transactions on, 2005. 17(7): 912-926.

[37] Xian, H. and D. Feng, Leakage Identification for Secret Relational Data Using Shadowed Watermarks, in International Conference on Communication Software and Networks. 2009. pp. 473-478.

[38] Guo, H., et al., A fragile watermarking scheme for detecting malicious modifications of database relations. Information Sciences, 2006. 176(10): 1350-1378.

[39] Guo, F., J. Wang, and D. Li. Fingerprinting relational databases. In: Proceedings of the 2006 ACM symposium on applied computing. 2006. ACM.

[40] Tsai, M.-H., et al. Fragile database watermarking for malicious tamper detection using support vector regression. in Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007. Third International Conference on. 2007. IEEE.

[41] Gupta, G., J. Pieprzyk, and M. Kankanhalli, Robust Numeric Set Watermarking: Numbers Don't Lie, in e-Business and Telecommunications. 2011, Springer. pp. 253-265.

[42] Gupta, G. and J. Pieprzyk, Database relation watermarking resilient against secondary watermarking attacks, in Information Systems Security. 2009, Springer. pp. 222-236.

[43] Gupta, G. and J. Pieprzyk. Reversible and blind database watermarking using difference expansion. In: Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop. 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[44] Jawad, K. and A. Khan, Genetic algorithm and difference expansion based reversible watermarking for relational databases. Journal of Systems and Software, 2013. 86(11): 2742-2753.

[45] Sun, J., Z. Cao, and Z. Hu, Multiple Watermarking Relational Databases Using Image. 2008: pp. 373-376.

[46] Wang, H., X. Cui, and Z. Cao, A Speech Based Algorithm for Watermarking Relational Databases. 2008: pp. 603-606.

[47] Sagar, R., Watermark based Copyright Protection for Relational database. International Journal of Computer Applications, 2013. 78(2).

[48] Agrawal, R., P.J. Haas, and J. Kiernan. A system for watermarking relational databases. in Proceedings of the 2003 ACM SIGMOD international conference on Management of data. 2003. ACM.

[49] Lafaye, J. An Analysis of Database Watermarking Security. 2007. pp. 462-467.

[50] Tian, J., Reversible data embedding using a difference expansion. IEEE Trans. Circuits Syst. Video Techn., 2003. 13(8): 890-896.

[51] Alattar, A.M., Reversible watermark using the difference expansion of a generalized integer transform. Image Processing, IEEE Transactions on, 2004. 13(8): 1147-1156.

[52] Xiao, X., X. Sun, and M. Chen, Second-LSB-Dependent Robust Watermarking for Relational Database. 2007: pp. 292-300.

[53] Kamran, M., Digital Right Protection of Relational Databases. 2012, National University of Computer and Emerging Sciences.

[54] Hamadou, A., *et al*., A Weight-based Semi-Fragile Watermarking Scheme for Integrity Verification of Relational Data. International Journal of Digital Content Technology and its Applications, 2011. 5(8): 148-157.

[55] Bender, W., *et al.,* Techniques for data hiding. IBM systems journal, 1996. 35(3.4): 313-336.

[56] Zhou, X., M. Huang, and Z. Peng. An additive-attack-proof watermarking mechanism for databases' copyrights protection using image. In Proceedings of the 2007 ACM symposium on applied computing. 2007. ACM.

[57] Wang, C., et al., ATBaM: An Arnold Transform Based Method on Watermarking Relational Data, in International Conference on Multimedia and Ubiquitous Engineering. 2008. pp. 263-270.

[58] Hu, Z., Z. Cao and J. Sun, An Image Based Algorithm for Watermarking Relational Databases. 2009: pp. 425-428.

[59] Huang, M., *et al*. A new watermark mechanism for relational data. In: Computer and Information Technology, International Conference on. 2004. IEEE Computer Society.

[60] Huang, K., *et al.*, A Cluster-Based Watermarking Technique for Relational Database. First International Workshop on Database Technology and Applications, 2009.

[61] Hu, T.-L., *et al.,* Garwm: Towards a generalized and adaptive watermark scheme for relational data, in Advances in Web-Age Information Management. 2005, Springer. pp. 380-391.

[62] Li, Y., H. Guo and S. Jajodia. Tamper detection and localization for categorical data using fragile watermarks. In: Proceedings of the 4th ACM workshop on Digital rights management. 2004. ACM.

[63] Bhattacharya, S. and A. Cortesi. Database Authentication by Distortion Free Watermarking. in ICSOFT (1). 2010. Citeseer.

[64] Bhattacharya, S. and A. Cortesi, Distortion-Free Authentication Watermarking, in Software and Data Technologies. 2013, Springer. pp. 205-219.

[65] Bhattacharya, S. and A. Cortesi, A generic distortion free watermarking technique for relational databases, in Information Systems Security. 2009, Springer. pp. 252-264.

[66] Prasannakumari, V., A robust tamperproof watermarking for data integrity in relational databases. Research Journal of Information Technology, 2009. 1(3): 115-121.

[67] Zawawi, N., *et al*., A novel watermarking approach for data integrity and non-repudiation in rational databases, in Advanced Machine Learning Technologies and Applications. 2012, Springer. pp. 532-542.

[68] Pournaghshband, V. A new watermarking approach for relational data. In: Proceedings of the 46th Annual Southeast Regional Conference on XX. 2008. ACM.